Splunk - Enterprise Installation Instructions

Last edited: Just now

Document Purpose

This document will outline all the steps required to successfully install Splunk Enterprise on RedHat Linux.

PreRequisite/Custom Settings

The following items are required before starting your server build.

- A server provisioned with necessary resources (CPU/memory/disk)
- A splunk ID on the server
- Sudo access to splunk ID
- Splunk installation .tgz in /tmp directory of server
- Disable Transparent Huge Pages:

    1. **sudo vi /etc/default/grub**
    2. Add **transparent_hugepage=never** to GRUB_CMDLINE_LINUX and save

    Example of contents before update:
    GRUB_TIMEOUT=3
    GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
    GRUB_DEFAULT=saved
    GRUB_DISABLE_SUBMENU=true
    GRUB_TERMINAL_OUTPUT="console"
    GRUB_CMDLINE_LINUX="nofb vmwgfx.enable_fbdev=0 crashkernel=auto rd.lvm.lv=vg_root/lv_root rd.lvm.lv=vg_root
    /lv_swap biosdevname=0 net.ifnames=0 audit=1"
    GRUB_DISABLE_RECOVERY="true"

Example of contents after update:
GRUB_TIMEOUT=3
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="nofb vmwgfx.enable_fbdev=0 crashkernel=auto rd.lvm.lv=vg_root/lv_root rd.lvm.lv=vg_root/lv_swap biosdevname=0 net.
ifnames=0 audit=1 transparent_hugepage=never"
GRUB_DISABLE_RECOVERY="true"

1. **sudo grub2-mkconfig -o /boot/grub2/grub.cfg**
2. **sudo reboot**

Remove Splunk Universal Forwarder Software

Base Server installation includes a copy of the Splunk Universal Forwarder Software and must be removed before the installation of the Enterprise version. Here are the commands to run to remove the sofware.

- sudo systemctl stop SplunkForwarder
- sudo systemctl disable SplunkForwarder
- sudo yum remove splunkforwarder.x86_64

- sudo rm /etc/systemd/system/SplunkForwarder.service

Install Software

---

1. Using Putty, SSH into server
2. cd /tmp
3. Install software to /apps:

   sudo tar xvzf splunk-8.1.0.1-24fd52428b5a-Linux-x86_64.tgz -C /apps

4. To keep a shared secret key, copy splunk.secret from /apps/splunk/etc/auth/splunk.secret from existing installation to new installation. Permissions are 400 (-r--------).
   Note: This was not done in the existing environment but is a best practice going forward.
5. Change owner of /apps/splunk:

sudo chown -R splunk:splunk /apps/splunk/

1. Start Splunk with additional arguments to enable boot start and start as user splunk(only for less than 7.3 versions):

sudo /apps/splunk/bin/splunk enable boot-start -user splunk --accept-license

for version after 7.3 run the following command instead of above :

sudo /apps/splunk/bin/splunk enable boot-start -systemd-managed 1 -user splunk --accept-license

1. When prompted, enter the local admin credentials (as stored in Cyber-Ark)
2. Now we need to edit the splunkd.service,

   sudo vi /etc/systemd/system/Splunkd.service
3. Once the file has loaded, add the following items in the “[Service] stanza:”,

Lines to add for versions less than 7.3 (No need to add any line from 7.3 version onwards just need to modify two lines mentioned below)
KillMode=mixed
KillSignal=SIGINT
TimeoutStopSec=10min

Example:
[Service]
Type=simple
Restart=always
ExecStart=/apps/splunk/bin/splunk _internal_launch_under_systemd
LimitNOFILE=65536
SuccessExitStatus=51 52
RestartPreventExitStatus=51
RestartForceExitStatus=52
KillMode=mixed
KillSignal=SIGINT
TimeoutStopSec=10min
User=splunk
Delegate=true
MemoryLimit=100G
CPUShares=1024
PermissionsStartOnly=true
ExecStartPost=/bin/bash -c "chown -R splunk:splunk /sys/fs/cgroup/cpu/system.slice/%n"
ExecStartPost=/bin/bash -c "chown -R splunk:splunk /sys/fs/cgroup/memory/system.slice/%n"

7.3 version Onward : No need to add any line just modify following two lines to match with existing indexers

from :
MemoryLimit=<it will be set with the memory of the host automatically>
TimeoutStopSec=360

To:
MemoryLimit=100G
TimeoutStopSec=10min

1. Save the file,

:wq

1. Verify service is enabled at startup:

```
sudo systemctl is-enabled Splunkd.service
```

1. Add splunk_home environment variable:

```
sudo vi /etc/profile.d/splunk_home.sh
```

Add the following:
```
#!/bin/sh
export SPLUNK_HOME=/apps/splunk
```

For indexers, also include the following line for the index storage:
```
export SPLUNK_DB=/splunkdb/
```

1. Save the file.

```
:wq
```

1. For License Server, Heavy Forwarders and SHC Deployers (not indexer cluster master or indexers, not search heads):
    1. Once you have logged into the server, switch to the "splunk" user,

        ```
        sudo -i -u splunk
        ```

    1. navigate to /apps/splunk/etc/system/local

        ```
        cd  /apps/splunk/etc/system/local
        ```

    1. create deploymentclient.conf

        ```
        touch deploymentclient.conf
        ```

    1. Edit the deploymentclient.conf file,

```
vi deploymentclient.conf
```

    1. Add the following lines to the deploymentclient.conf file,

```
[target-broker:deploymentServer]
targetUri = sssdsapp3.gwl.bz:8089
```

Notes:
For DMZ hosts, set targetUri to sssdsapp4.gwl.bz:8089
Deployment apps have been created to distribute additional configurations to the Splunk servers. Reference other documentation to manually configure search heads or indexers.

    1. Save the file,

```
:wq
```

1. For Search head cluster members, configure alert actions configuration (allows users to click results and be taken through the load balancer)

    ```
    cd /apps/splunk/etc/system/local/
    ```

1. Create the alert_actions.conf file,

    ```
    touch alert_actions.conf
    ```

1. Edit the alert_actions.conf file,

   vi alert_actions.conf

1. Enter the following details,

```
# This file configures global saved search actions.
#
# Set the hostname that is displayed in the link sent in alerts.
# The resulting link is "http://hostname:port/....."
# Can be any string, or empty to pick up the hostname automatically.
#
hostname=splunk.gwl.bz:443
[email]
# SMTP server sending out all alert emails
#
mailserver = mail-relay.gwl.bz
footer.text = If you believe you've received this email in error, please contact SMOMonitoring@gwl.ca.\
\
splunk > the engine for machine data
```

NOTE: For ITSI Search head members, set the above hostname=itsi.gwl.bz:443

1. Save the file,

   :wq

1. At the command prompt, enter “exit” to log out of the “Splunk” id.
2. For new Splunk Indexers

   sudo vi /apps/splunk/etc/splunk-launch.conf

   Add ----- SPLUNK_DB=/splunkdb
   :wq
   sudo chown splunk:splunk /apps/splunk/etc/splunk-launch.conf
   sudo chmod 644 /apps/splunk/etc/splunk-launch.conf

1. Reload daemon,

```
sudo systemctl daemon-reload
```

1. Start service,

```
sudo systemctl start Splunkd.service
```

1. From your browser, the web UI should now be available at http://<server>.gwl.bz:8000
   Note: Initial login can only be completed with credentials specified above

~~Date/Time Fix~~

~~As of December, 2019 on version 7.2.5.1 the datetime.xml file needs to be updated. The current configuration deploys an app to all deployment clients from the deployment server; to all search head cluster members from the deployer; and to indexer cluster members from the cluster master. If the Splunk installation needs the fix manually applied, be sure to add the appropriate fix from apps_date_patch_props_v3.zip to /etc/apps (idxc_date_patch_props to indexers; all_date_patch_props to all others).~~

~~This is not necessary on 7.2.9.1 and will be removed in version 8.0.1+~~

~~More information is available at https://docs.splunk.com/Documentation/Splunk/8.0.0/ReleaseNotes/FixDatetimexml2020~~

HTTPLIB2 Temporary Certificate Fix

An issue was identified in April, 2020 where Splunk add-ons using httplib2 generate temporary .crt files for SSL communication, but do not remove them after. Case 1716346 was opening with Splunk support; in the meantime the following should be implemented on any Splunk heavy forwarders executing add-ons that retrieve data in order to ensure the .crt files are deleted on an hourly basis:

1. Create a shell script: sudo vi /usr/local/bin/httplibcrtdelete.sh
2. Add the following: find /tmp/*.crt -mmin +60 -delete
3. Add the script to a cron job: sudo crontab -e
4. Add the following: 5 0-23 * * * /usr/local/bin/httplibcrtdelete.sh 1>/dev/null 2>&1